



Universidad Carlos III de Madrid

Escuela Politécnica Superior

Grado en Ingeniería Telemática

*Estudio experimental de mecanismos de traspaso
controlados por la red en entornos 802.11*

Trabajo Fin de Grado

Autor:	Rubén Montoro Terciado
Tutor:	Pablo Serrano Yáñez-Mingot
Director:	Carlos Jesús Bernardos Cano

Julio de 2013

Trabajo Fin de Grado

Estudio experimental de mecanismos de traspaso controlados por la red en
entornos 802.11

Autor

Rubén Montoro Terciado

Tutor

Dr. Pablo Serrano Yáñez-Mingot

Director

Dr. Carlos Jesús Bernardos Cano

Realizado el acto de defensa y lectura del Trabajo Fin de Grado el día de
Septiembre de 2012 en Leganés, en la Escuela Politécnica Superior de la Universidad
Carlos III de Madrid, el tribunal:

PRESIDENTE:

SECRETARIO:

VOCAL:

acuerdan otorgarle la calificación de:

CALIFICACIÓN:

Leganés, a de Julio de 2013

A mis padres, a mi familia, a Lara y a mis amigos

Agradecimientos

En primer lugar, dar las gracias a mis padres, Miguel Ángel y Piedad. Sin ellos, este trabajo y lo que conlleva no habría sido posible. Dar las gracias también a mi hermano Adrián, a toda mi familia, y a Charo, que me han apoyado incondicionalmente desde el principio.

En segundo lugar, dar las gracias a mis tutores: Pablo y Carlos. Gracias por su ayuda, su apoyo y su trabajo a lo largo de este último año de carrera. Espero que se hayan dado cuenta de que Mac no funciona bien y de que Windows va genial... a veces.

Dar las gracias también a mis compañeros de universidad: Adrián, Daniel, Laura, Raul, Yago y Amelia. Hemos pasado buenos ratos juntos, aunque fuera un sábado a las 10.00 en la biblioteca haciendo trabajos o después de largas prácticas en las que después de una hora en el laboratorio, no funcionaba nada.

También quiero dar las gracias a una persona muy especial que ha estado conmigo desde que empezó mi pequeño infierno al que algunos llaman universidad. Una persona que me ha dado una colleja cuando estaba muy nervioso antes de un examen y que ha sido la primera en felicitarme cuando he aprobado. Por eso y por mucho más, gracias Lara.

Y por último, y no menos importante, mis amigos. Gracias por las tardes de cervecero y cenas para relajarnos después de una dura semana de universidad. Guille, Ibeth, Duri, Alex, Rubén, Tania, Nacho, etc. Gracias por todo.

La vida no es fácil, para ninguno de nosotros. Pero... ¡qué importa!
Hay que perseverar y, sobre todo, tener confianza en uno mismo.
Hay que sentirse dotado para realizar alguna cosa y que esa cosa
hay que alcanzarla, cueste lo que cueste.
Marie Curie (1867-1934)

Resumen

En los últimos años, el desarrollo de la tecnología de acceso a la red inalámbrica se ha incrementado de manera exponencial, permitiendo a una gran variedad de nodos (*smartphones*, ordenadores portátiles, modems-USB) acceder a Internet desde cualquier lugar y en cualquier momento. Sin embargo, debido a este crecimiento exponencial, las redes de telecomunicación empiezan a estar saturadas, con lo cual es necesario implementar mecanismos que permitan distribuir el tráfico generado por dichos nodos entre los puntos de acceso que estén en su alcance.

Las futuras redes deberán ser de alta densidad, formadas por multitud de puntos de acceso muy cercanos los unos a los otros, lo cual permitirá absorber un mayor tráfico de los nodos que se conecten a nuestra red. Por otro lado, al incrementarse el número de usuarios en la red será necesario desarrollar mecanismos que permitan gestionar de manera sencilla tanto el tráfico que fluya por la red como la gestión de los usuarios conectados a la misma con el fin de balancear la carga de trabajo de una manera eficaz.

En este Trabajo de Fin de Grado se presenta un estudio mediante el cual una red 802.11 sin soporte en el terminal puede forzar el traspaso de un nodo conectado a un punto de acceso a otro mediante tres simples mecanismos. Para ello mediremos el tiempo de traspaso de nodos con diferentes sistemas operativos desde que deja de recibir tráfico hasta que vuelve a recibirlo. Nuestro objetivo es medir dichos tiempos para caracterizar cómo afecta a las prestaciones.

En la primera fase, se ha desplegado una pequeña infraestructura inalámbrica en los laboratorios del Departamento de Ingeniería Telemática de la Universidad Carlos III de Madrid con puntos de acceso Fonera 2.0n y Linksys WRT54GL, ambos con firmware basado en Linux. En la segunda fase se han realizado pruebas automatizadas de mecanismos para gestionar los nodos conectados a una red, dichos nodos funcionaban con Windows8, Mac OS X y Ubuntu. Tras esto, se han procesado los datos obtenidos y se han obtenido unas conclusiones de cual es el modo más eficaz de realizar el traspaso. A la vista de los resultados obtenidos, se concluye que la duración de un traspaso varía enormemente en función del mecanismo que utilizemos y el sistema operativo.

Abstract

In recent years, there has been an exponential growth in the traffic using wireless network access, with a variety of nodes (smartphones, laptops, USB-modems) that access the Internet from anywhere and at any time. However, due to this exponential growth, telecommunication networks begin to be saturated, and therefore there is the need to develop mechanisms in order to distribute nodes among the various access points in range.

It is envisioned that future networks will be very networks, with a large number of access points deployed very close to each other, in order to provide access to the large number of nodes. With this increase in density and the number of nodes, there is the need to develop mechanisms to manage the way in which users are connected to the network, in order to efficiently balance the workload.

In this Project, we study three representative mechanisms that can be used in an 802.11 network to force the re-association of a node to a different access point, with no additional support from the nodes. We measure the amount of time the node is disconnected from the network, and analyze the impact of the mechanism and operating system on performance.

In a first phase, we have deployed a small test-bed in a laboratory at the Department of Telematic Engineering, composed of Fonera 2.0n and Linksys WRT54GL nodes, both using Linux-based distributions. In a second phase, we have developed the required code for automated testing of the mechanisms. Then, we have processed the data and analyzed the results, determining what is the most effective configuration to perform the re-association. Based on the results, we conclude that the duration of the re-association process heavily depends on the mechanisms used and operating system.

Índice General

Índice General	xv
1. Introducción	1
1.1. Motivación	1
1.2. Objetivo	1
1.3. Estructura de la memoria	2
2. Mecanismos de traspaso estudiados	3
2.1. Desconexión abrupta del punto de acceso	3
2.2. Mecanismo de traspaso basado en RSSI	3
2.3. Desautenticación del terminal	4
2.4. Definición de traspaso	4
3. Descripción del despliegue y metodología	7
3.1. Despliegue	7
3.1.1. Hardware	9
3.1.2. Software	9
3.2. Metodología	12
4. Resultados obtenidos	17
4.1. Impacto del mecanismo de traspaso	17
4.2. Impacto del sistema operativo	19
4.3. Análisis de los mecanismos de traspaso	21
4.3.1. Mecanismo por traspaso abrupto	21
4.3.2. Mecanismo de desautenticación	22
4.3.3. Mecanismo de reducción de potencia	22
4.4. Redes con diferente SSID	30
4.5. Resumen	31
5. Conclusión y trabajos futuros	33
A. Planificación de tareas y presupuesto	35
A.1. Descomposición en tareas	35
A.2. Planificación con el diagrama de fases de ejecución detallado	38
A.3. Recursos	40
A.4. Presupuesto de Proyecto	40
B. Instalación y configuración de los routers inalámbricos	43

B.1. Instalación de OpenWRT en Fonera 2.0N	43
B.2. Instalación de OpenWRT en Linksys WRT54GL	45
Bibliografía	49

Capítulo 1

Introducción

1.1. Motivación

En la actualidad, las redes que conforman Internet son un vehículo que ayuda al hombre a desarrollar su trabajo con una alta eficacia. Desde el nacimiento de la red hasta ahora, la cantidad de equipos que pueden conectarse a la red, y por tanto, de usuarios ha crecido de manera exponencial. Según las previsiones que desvela un estudio de Cisco[Cis13], el tráfico de datos móviles se multiplicará por 13 desde 2012 a 2017. Esto conlleva que el diseño de las redes del futuro tenga como problema la alta densidad de usuarios a los que debe dar capacidad.

Por tanto, una de las soluciones más factibles para aumentar la capacidad de la red es aumentar la densidad de puntos de acceso para soportar el aumento de la demanda de tráfico por parte de los usuarios. Para poder optimizar nuestros recursos, es conveniente evitar la sobrecarga en las redes mediante el traspaso de equipos de un punto a otro de la red con el fin de balancear la carga que se genera en la red y optimizar las prestaciones. En nuestro caso, las redes inalámbricas, podremos distinguir dos tipos de traspaso:

- Traspasos iniciados por el nodo: el objeto móvil decide a que punto de acceso de la red se conecta. La ventaja es que el nodo podrá adaptarse a aquello le garantice mejores prestaciones. Sin embargo, esto requeriría de un soporte para poder utilizar la red, es decir, se debe desarrollar más complejidad en el nodo con el fin de que pueda ponerse de acuerdo con la red para ubicarse en el punto que más le convenga.
- Traspasos iniciados por la red: gracias a la visión global de la red, se puede optimizar los recursos y garantizar las mejores prestaciones en toda ella. La red trata a todos los nodos por igual, con lo cual, no es necesario añadir un soporte de red al nodo. Sin embargo, los terminales son heterogéneos, es decir, no son iguales los unos a los otros y por tanto, no tienen porque comportarse igual ante la gestión de la red.

1.2. Objetivo

El objetivo de este Trabajo de Fin de Grado es caracterizar el tiempo de traspaso de diferentes sistemas operativos en redes 802.11 sin soporte en el terminal. Para ello, vamos a realizar las siguientes tareas:

- Estudio de los mecanismos existentes para realizar traspasos en redes 802.11 sin soporte del terminal.
- Implementación de un entorno basado en firmware Linux.
- Desarrollo de código para automatizar pruebas.
- Pruebas del comportamiento de distintos sistemas operativos ante distintos mecanismos de traspaso.
- Sintetización de resultados.

1.3. Estructura de la memoria

- Capítulo 1. Introducción.
- Capítulo 2. Mecanismos de traspaso.
- Capítulo 3. Despliegue experimental y metodología.
- Capítulo 4. Resultados obtenidos.
- Capítulo 5. Conclusiones y trabajo futuro.
- Apéndice A. Planificación de tareas y presupuesto.
- Apéndice B: Instalación de OpenWRT en los nodos inalámbricos.

Capítulo 2

Mecanismos de traspaso estudiados

En este capítulo hablaremos sobre los algunos de los mecanismos más representativos que existen para hacer traspasos iniciados por la red sin soporte en el terminal. Cuando un nodo móvil pierde la cobertura con el punto de acceso al que estaba conectado, es lógico que se conecte a otro que este en su área de cobertura. Para ello, utilizamos distintos mecanismos que son empleados por la red.

Antes de explicar los diferentes mecanismos, definiremos el concepto de network manager, que es el software que se encarga de la gestión de las interfaces de red. En lo que a nosotros se refiere, dicho software se encarga de mantener la conexión con los puntos de acceso a los que se halla conectado el ordenador y en caso de perder la conexión con un punto de acceso, reaccionar ante los mecanismos de traspaso a los que se le somete y autenticarse en un nuevo router. Los network manager que se han utilizado en esta memoria no disponen de soporte por parte de la red, lo cual reduce la complejidad en el terminal.

2.1. Desconexión abrupta del punto de acceso

Este mecanismo es el más simple de los que vamos a estudiar en este Trabajo de Fin de Grado. Consiste desactivar el punto de acceso al que se encuentra conectado un nodo. De esta manera, los nodos deberán escanear el medio a su alrededor para conectarse a otro punto de acceso. Una de las ventajas de este método es que su escalabilidad es muy alta, puesto que no es necesario alertar a los nodos conectados de que el mecanismo va a ser ejecutado. Otra de las ventajas es que su eficacia es del 100 % pues siempre funciona. La duración del traspaso está condicionada por el tiempo que tarde el nodo en detectar que el punto de acceso no está operativo, el tiempo que tarde en efectuar el escaneo de redes a su alrededor y el tiempo que tarde en asociarse a otro punto de acceso.

2.2. Mecanismo de traspaso basado en RSSI

Este mecanismo de traspaso es similar al anterior pues se basa en el control de la potencia de transmisión de la antena del punto de acceso. Mientras que en el caso anterior, la potencia de transmisión se reduce inmediatamente, en este caso dicha reducción es gradual. Mediante la calidad y la potencia de la señal recibida, el ordenador comprobará

si existen puntos de acceso a su alrededor a los que conectarse para poder obtener mejor calidad de acceso a la red a través de otro punto de acceso.

Es difícil detectar el umbral en el que se debe accionar este mecanismo porque hay que evitar iniciar el traspaso con gran retraso. Si no, la conexión con la red a la que actualmente estamos conectados se perderá totalmente. Para determinar este umbral, este proceso se realiza mediante un promediado de los ecos que se producen en el entorno del nodo móvil, evitando así el efecto ping-pong (efecto indeseado del cambio de estación base innecesariamente).

2.3. Desautenticación del terminal

Este mecanismo es usado por la red para forzar la desconexión de un usuario mediante la desautenticación del nodo en un punto de acceso. Por ejemplo, si quisiéramos que un nodo móvil que está anclado a un punto de acceso de nuestra red se conectara a otro por el motivo que fuera (saturación de la red, seguridad, etc.) podríamos forzarlo mediante este mecanismo, tan sencillo como mandar un mensaje de desautenticación desde el router al nodo móvil. Una de las desventajas de este método es que es algo menos escalable y más complejo, ya que para cada nodo que queramos desasociar debemos mandar una trama de desautenticación. Otra de las desventajas es que la duración de un traspaso mediante este mecanismo esta condicionada en función de la persistencia del network manager por permanecer conectado a una red de la que es desautenticado. Sin embargo, como veremos en los siguientes capítulos, es el método más eficaz para realizar un traspaso.

2.4. Definición de traspaso

En el campo de las redes inalámbricas existen múltiples interpretaciones sobre qué es un traspaso. De cara a explicar los resultados de esta memoria, nuestro concepto de traspaso será el periodo de tiempo que transcurre entre la recepción del paquete con datos validos para el usuario que se recibe antes de iniciar el mecanismo de traspaso y el primer paquete que se recibe tras restablecer la conexión tras asociarnos a otro punto de acceso. Podemos ver una definición gráfica de este concepto en la gráfica [2.1](#).

En dicha gráfica podemos ver un flujo de paquetes en sentido descendente, desde el router hacia el nodo móvil. Consideraremos la duración del traspaso la diferencia de tiempo que transcurre entre dos paquetes de datos con valor para el usuario. En este intervalo de tiempo, el network manager del ordenador habrá reaccionado ante un mecanismo de traspaso y habrá realizado las gestiones necesarias para conectarse a otro punto de acceso.

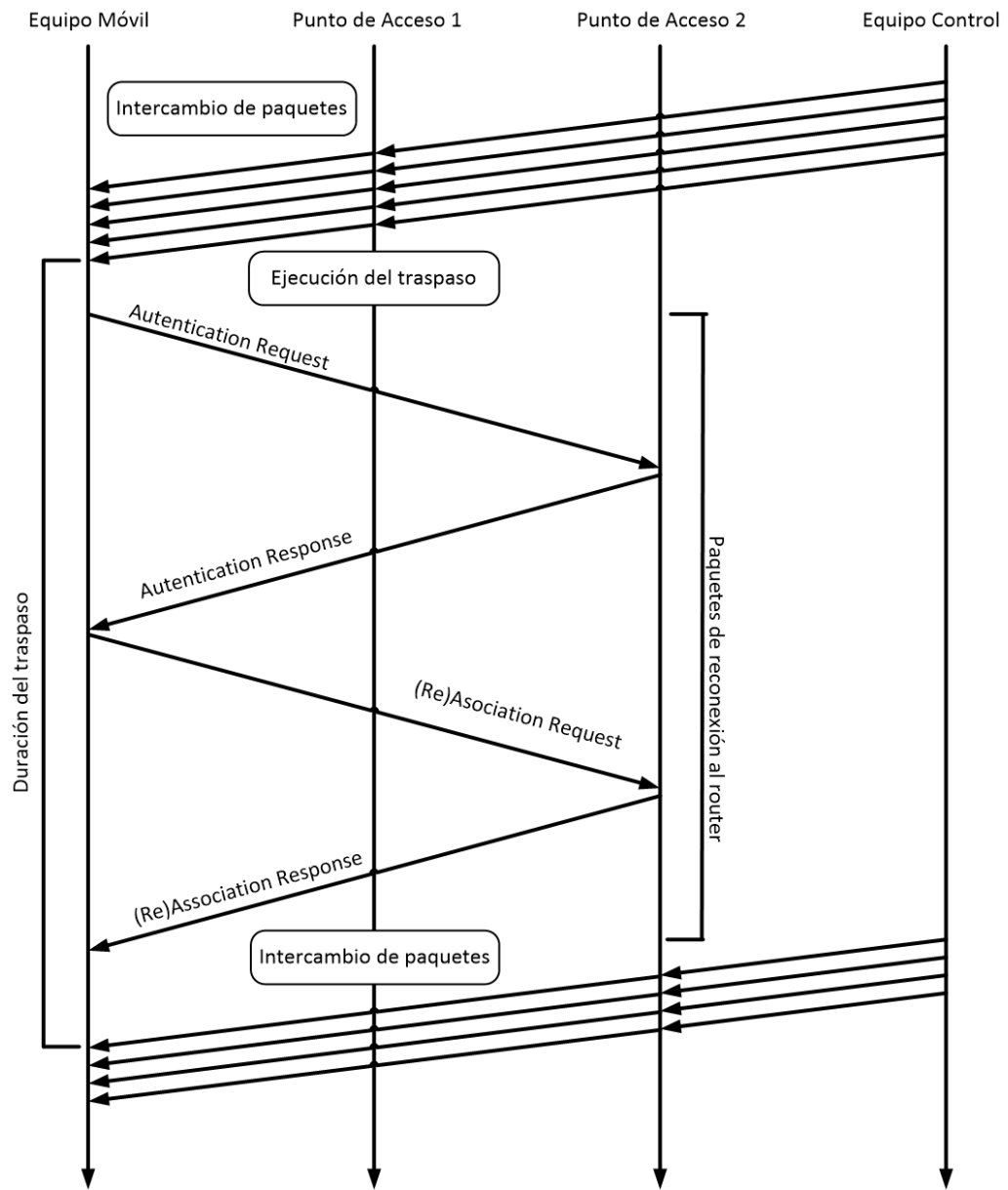


Figura 2.1: Escenario para definición de un traspaso

Capítulo 3

Descripción del despliegue y metodología

En este apartado describiremos el entorno de trabajo en el que hemos realizado las pruebas así como la metodología que se ha seguido para obtener los resultados que mostramos en el siguiente capítulo de esta memoria.

3.1. Despliegue

Para desarrollar este Trabajo de Fin de Grado se ha utilizado el entorno descrito en la gráfica 3.1

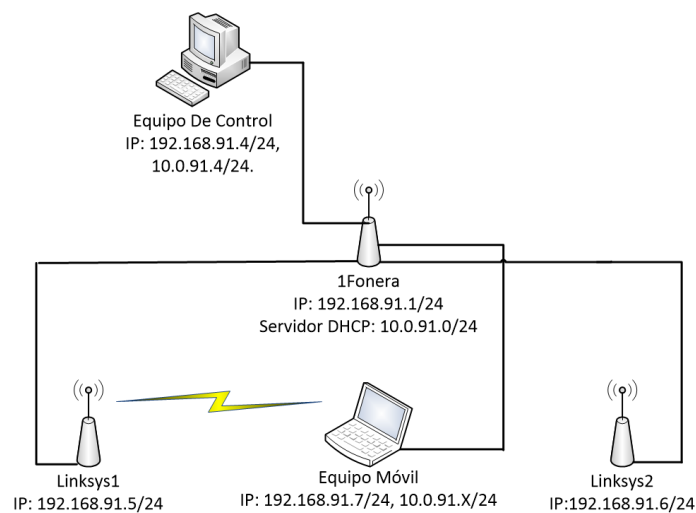


Figura 3.1: Entorno de desarrollo

Podemos observar que se trata de una red con topología de árbol con dos ramas. El router 1Fonera alberga un servidor DHCP que asigna las direcciones IP a los nodos que se conecten a cualquier punto de la red. Así, al realizar traspasos, la dirección IP no cambiará (el traspaso no es transparente al nivel de transporte), aumentando la velocidad del traspaso. 1Fonera será el router que tenga conexión con el ordenador de gestión y con Internet (en caso de que algún nodo de la red necesite acceder a recursos externos). En las

imágenes 3.2, 3.3 y 3.4 podemos observar la disposición real de los routers en el laboratorio donde hemos realizado las pruebas

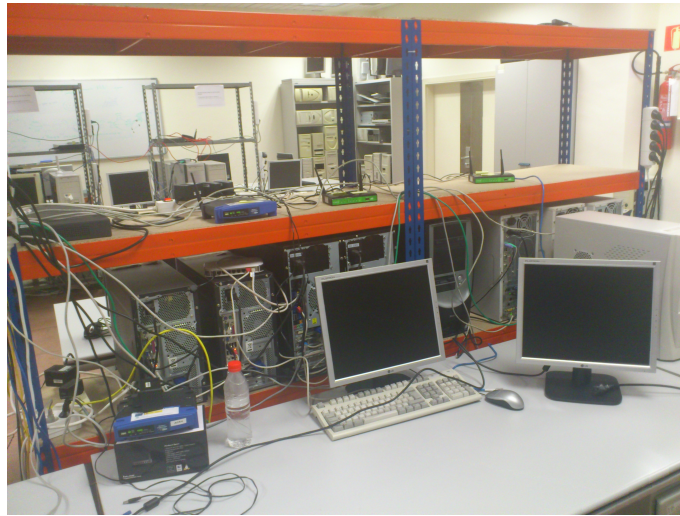


Figura 3.2: Entorno de trabajo



Figura 3.3: Perfil del entorno de trabajo



Figura 3.4: Router Linksys2 bajo suelo cubierto de aislante en el laboratorio

3.1.1. Hardware

El entorno de desarrollo está compuesto dos routers Linksys WRT54GL [Tabla 3.2] y un router Fonera 2.0N [Tabla 3.1]. Hemos elegido estos modelos porque permiten la instalación de un firmware basado en Linux, que permite al usuario controlar todas las capacidades del router, lo cual es necesario para el desarrollo de este trabajo. En las siguientes tablas se resumen las principales características de los puntos de acceso.

También utilizaremos tres ordenadores para realizar pruebas: uno de ellos será el ordenador de control [Tabla 3.3], desde el cual realizaremos las pruebas, otro servirá de nódo móvil y funcionará con Windows 8 Profesional y Ubuntu 12.10 [Tabla 3.4] y el último también será un nodo móvil que funcionará con Mac OS X [Tabla 3.5]. Podemos consultar las características detalladas de todo el hardware descrito en esta sección en las tablas que se adjuntan en este capítulo.

3.1.2. Software

Para poder medir con precisión el tiempo exacto de un traspaso, usamos la aplicación MGEN, la cual es un generador de tráfico que permite obtener en el nodo que recibe tráfico una salida textual y ordenada de los paquetes que ha recibido. Tras analizar esta salida podemos determinar el tiempo que pasa entre que el receptor recibe dos paquetes con una

Característica	Descripción
CPU	Ralink 300 Mhz
RAM	64 MB
Flash	8 MB
Dimensiones	30 mm x 157 mm x 127 mm (excluyendo la antena)
Puerto USB	Ralink USB 2.0/1.1
Puertos Ethernet	1 puerto Ethernet WAN (10/100Mbps) para conexión a Internet. 4 puertos Ethernet LAN (10/100Mbps)
Antenas	2 antenas dipolares fijas a 3dBi (Tecnología 2T2R MIMO)
Autenticación	WEP 64bit/128 bit, WPA, WPA2, WPA mixto
Encriptación	TKIP, AES, Mixto
Soportes de red	IEEE 802.11g/b and 802.11n (superior a 150 Mbps)
SSIDs	Una pública (llamada FON_FREE_INTERNET por defecto), no encriptado y una privada (llamada MyPlace por defecto) encriptado con WPA-PSK

Tabla 3.1: Características técnicas de router Fonera 2.0n

Característica	Descripción
CPU	Broadcom BCM5352 200 MHz
RAM	16 MB
Flash	4 MB
Dimensiones	186mm x 48mm x 154 mm (excluyendo la antena)
Puertos Ethernet	1 puerto Ethernet WAN (10/100Mbps) para conexión a Internet. 4 puertos Ethernet LAN (10/100Mbps) Switch
Antenas	2
Autenticación	WEP, WPA, WPA2, WPA mixto
Encriptación	TKIP, AES, Mixto
Soportes de red	IEEE 802.11g/b

Tabla 3.2: Características técnicas de router Linksys WRT54GL

Característica	Descripción
Procesador	Intel(R) Core(TM)2 Duo CPU E6750 2.66 Ghz
Memoria RAM	2GB
Disco Duro	320 GB
Sistema Operativo	Ubuntu 10.10
Ethernet	Gigabit 100Mbps/1Gbps
Wireless	802.11 a/b/g/n

Tabla 3.3: Características técnicas del ordenador de control

Característica	Descripción
Procesador	Intel Core i7-2630QM 2Ghz
Memoria RAM	4GB
Disco Duro	500 GB
Sistema Operativo	Windows 8 Profesional / Ubuntu 12.10
Ethernet	Gigabit 100Mbps/1Gbps
Wireless	Intel Centrino Wireless N100 802.11 a/b/g/n

Tabla 3.4: Características técnicas del nodos móvil con Windows 8 y Ubuntu

Característica	Descripción
Procesador	Intel Core 2 Duo 2.2Ghz
Memoria RAM	4GB 667 Mhz DDR2 SDRAM
Disco Duro	120 GB
Sistema Operativo	Mac OS X 10.6.8
Ethernet	Gigabit 100Mbps/1Gbps
Wireless	Atheros 802.11 a/b/g/n

Tabla 3.5: Características técnicas del nodo móvil con Mac OS X

precisión de millonésima de segundo. Además, podemos determinar que paquetes se han perdido gracias a un campo id que se encuentra en la carga del paquete. Para garantizar que la medida del tiempo es lo más precisa posible, todos los equipos cuentan con un cliente NTP que actualiza la hora constantemente para evitar una desincronización de los equipos.

No obstante, este programa solo es aplicable en caso de realizar las pruebas con Ubuntu y con Mac Os X, ya que ambos poseen una arquitectura de tipo Linux. Sin embargo, MGEN no es compatible con una versión de 64 bits de Windows 8 Professional, por tanto, hemos realizado estas pruebas utilizando un generador de tráfico diferente. iPerf es un generador de tráfico, similar a MGEN, que permite conocer el orden en el que se generan los paquetes al campo identificación de la cabecera IP.

Para capturar tanto el tráfico generado utilizaremos la herramienta de captura Wireshark, la cual nos permite comprobar y capturar los paquetes que se han intercambiado durante un experimento. Por último, hemos usado Matlab para procesar los resultados obtenidos y obtener gráficas fácilmente comprensibles.

Para terminar, hemos podido verificar la versatilidad del software libre (OpenWRT, Ubuntu, etc.) ya que, aparte de reducir el impacto económico de un proyecto drásticamente, en muchas ocasiones permite realizar operaciones que el software que provee el fabricante de un hardware, e incluso, aumentar su utilidad.

3.2. Metodología

En cada experimento realizado en este Trabajo de Fin de Grado hemos realizado 100 pruebas para obtener una buena caracterización de cada mecanismo. Dichos experimentos han sido realizados a diversas horas del día, sin que influya en ellos el tráfico que existe en el medio que nos ha rodeado. Cada experimento ha tenido una duración aproximada de dos horas y media.

Para realizar los experimentos, es necesario que estos estén automatizados, pues resulta muy complejo realizar 100 pruebas manualmente. Para ello, hemos desarrollado un código escrito en bash de Linux, el cual puede ser consultado¹, que simplifica la ejecución de las pruebas en el ordenador de control, el cual funciona con Ubuntu. Dicho código varía mínimamente a la hora de realizar pruebas con los diferentes sistemas operativos, tan solo es necesario cambiar las rutas de acceso a las diferentes máquinas. Podemos consultar el diagrama de flujo de dicho código en la figura 3.5.

Gracias a las herramientas descritas anteriormente podemos medir con precisión la duración de los métodos de traspasos. Mediante los generadores de tráfico que hemos comentado en la sección 3.1.2, creamos un flujo de tráfico UDP desde el ordenador de control hasta el nodo móvil que hemos conectado a la red (basaremos siempre los experimentos en un sentido descendente). Empezamos a medir la duración del traspaso en el momento en el que ordenador recibe el último paquete UDP. Tras esto, y pasado el tiempo de traspaso, aun sin determinar, el flujo de datos vuelve a restablecerse, pero esta vez a través de otro punto de acceso.

Una vez que hemos concluido un experimento y obtenemos trazas con los resultados que indican qué paquetes se han perdido, hay que conocer cuántos paquetes se han perdido y cuanto tiempo ha tardado el equipo móvil en restablecer la conexión con la red inalámbrica.

¹<https://www.dropbox.com/s/75bilgrt85hea4h/C%C3%B3digoTFG.zip>

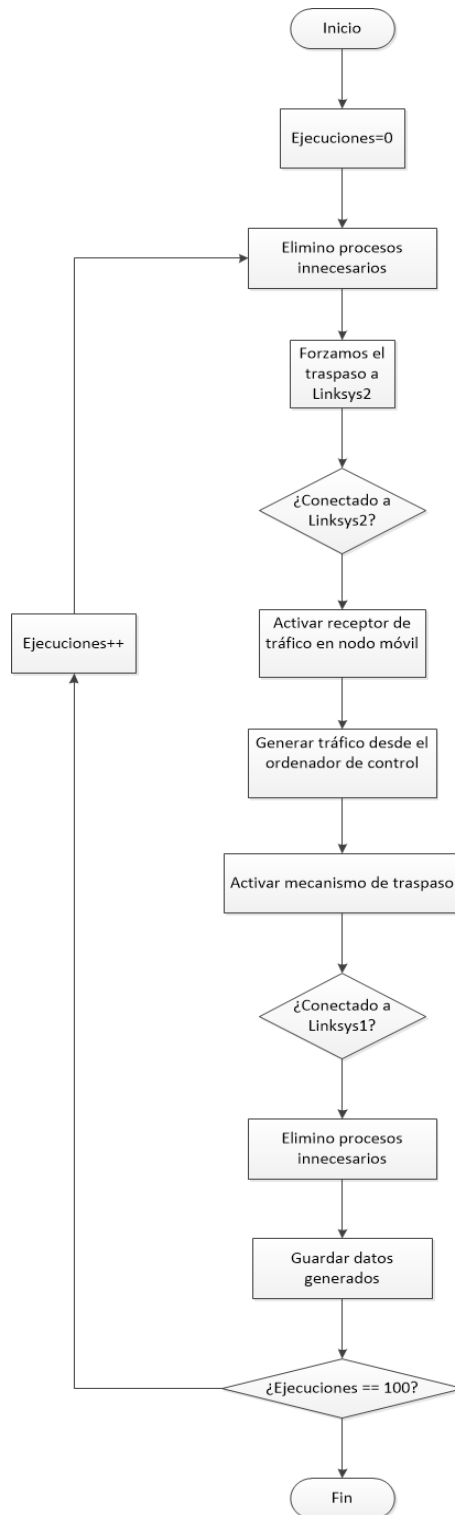


Figura 3.5: Diagramas de flujo del código desarrollado en bash

Para ello, utilizaremos un código Java que hemos desarrollado. Como dato, indicar que para obtener una información procesable en el caso de Windows 8 es necesario usar el capturador WinDump, el cual transformará la información obtenida con iPerf en archivos de texto entendibles por el programa Java antes mencionado.

Dicho código Java lee paquete por paquete, guardando en una variable el instante del

tiempo en que se ha recibido. Si la identificación de ese paquete no es consecutiva a la del anterior, se calcula el tiempo que ha pasado entre el paquete recibido y el anterior y se guarda en un fichero de texto que posteriormente será procesado por Matlab. Esto se repite con los múltiples archivos que contienen trazas y que se generan durante los experimentos. Como se indica anteriormente, Windows 8 produce archivos ligeramente distintos, con lo cual ha sido necesario adaptar el código a las trazas producidas en Windows 8 con iPerf. Podemos consultar el diagrama de flujo en la gráfica 3.6 y su correspondiente código².

Tras el paso anterior, se procede a procesar los archivos generados por el código Java con Matlab. Dado que al ser una transmisión UDP se pueden perder más paquetes aparte de los que se han perdido durante el traspaso (UDP no garantiza la entrega de todos los paquetes enviados), el código Matlab selecciona el valor de tiempo perdido más alto y lo representa junto a otros valores en la CDF. En el siguiente capítulo se explica los resultados de estos experimentos.

²<https://www.dropbox.com/s/75bilgrt85hea4h/C%C3%B3digoTFG.zip>

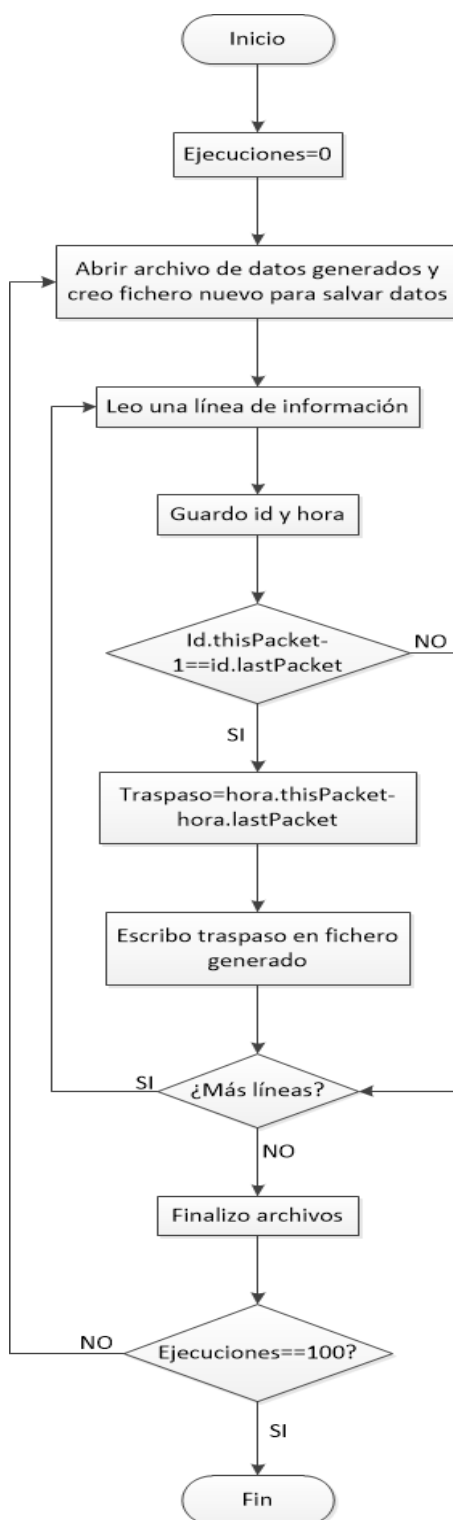


Figura 3.6: Diagramas de flujo del código desarrollado en Java

Capítulo 4

Resultados obtenidos

En esta sección mostraremos los resultados obtenidos en la realización de las pruebas anteriormente descritas con los sistemas operativos Windows 8, Mac OS X y Ubuntu atendiendo a tres ejes de estudio:

- Sistema operativo: la correcta programación del network manager de una distribución puede ser una clave fundamental en la duración de un traspaso. Por consiguiente, cada network manager reacciona de manera diferente ante cada tipo de traspaso
- Canal: la interferencia que se produce en los canales del estándar 802.11 es un factor importante ya que podemos evitar colisión de paquetes si operamos en redes de distinto canal aunque sean del mismo SSID. Para comprobar la influencia de este factor, las pruebas se han realizado con routers tanto el el mismo como en diferente canal.
- Tipo de traspaso: cada mecanismo hace reaccionar de manera diferente al network manager de los nodos móviles. Comprobaremos en las siguientes secciones que cada método tiene sus ventajas (rapidez en el traspaso, por ejemplo) y sus inconvenientes (escalabilidad).

4.1. Impacto del mecanismo de traspaso

Para comenzar, vamos a estudiar el impacto de los mecanismos en cada sistema operativo. Podemos ver las funciones de distribución acumulada (CDF en adelante) de cada distribución en las gráficas [4.1](#), [4.2](#) y [4.3](#).

En primer lugar, la gráfica [4.1](#) indica que el método que mayor eficacia ha demostrado durante las pruebas para realizar un traspaso en Mac OS X es mediante mensajes de desautenticación que envía el router al nodo para forzar que se conecte a otro nodo. Los otros dos métodos son más lentos e ineficaces dado que afectan no solo al nodo al que se quiere expulsar, sino a todos los nodos que estén vinculados a ese router. Sin embargo, el mecanismo que ha presentado los peores resultados ha sido el traspaso abrupto, donde podemos apreciar una fuerte interferencia de la variable canal. Se observa que en un traspaso en mismo canal, en el 50 % de los casos, la duración máxima es de 15 segundos mientras que en las mismas condiciones en igual canal, la duración máxima solo alcanza los 10 segundos.

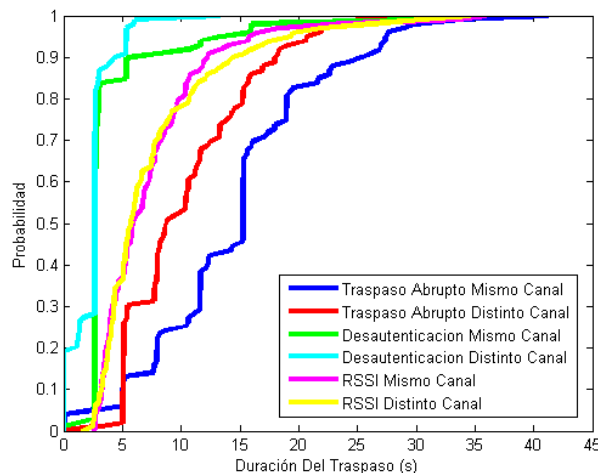


Figura 4.1: CDF de mecanismos de traspaso en Mac OS X

Observando la gráfica 4.2, que contiene el resultado de las pruebas realizadas con Ubuntu, podemos concluir que, como en el caso anterior, el network manager de Ubuntu reacciona favorablemente al mecanismo de envío de una trama de desautenticación, indistintamente de que los puntos de acceso se encuentren en el mismo o en diferente canal. También podemos observar que en el resto de los métodos de traspaso, la influencia de la variable canal es despreciable. El mecanismo que peores ha obtenido ha sido el traspaso por reducción de potencia, pues como apreciamos en la gráfica 4.2, la duración máxima de un traspaso es de 35 segundos en el 90 % de los casos. Se puede apreciar que la velocidad del traspaso es mucho más suave con este mecanismo que con los otros dos.

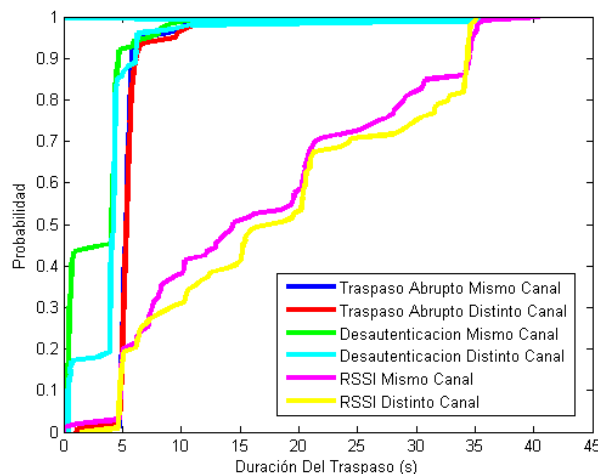


Figura 4.2: CDF de mecanismos de traspaso en Ubuntu

Por último, en la gráfica 4.3 se representa la CDF que se ha producido tras repetir las pruebas anteriormente descritas con Windows 8. Lo primero que podemos apreciar es una reducción significativa de la duración máxima de los traspasos (note la reducción del eje del dibujo). Una vez más, se comprueba que el método que mejor resultados ha producido ha sido forzar una trama de desautenticación (independientemente del canal en el que se encuentren los puntos de acceso), seguido del mecanismo por traspaso abrupto. En el 100 % de los casos, la duración máxima de los traspasos es de 0.9 y 2 segundos, respectivamente.

Por último, si comparamos el mecanismo de reducción de potencia con los otros dos sistemas operativos, podemos ver que ha reaccionado más rápidamente a pesar de no ser tan efectivo en Windows 8 como el traspaso abrupto y la desautenticación. No obstante, la duración máxima en el 100 % de los casos es de 6 segundos.

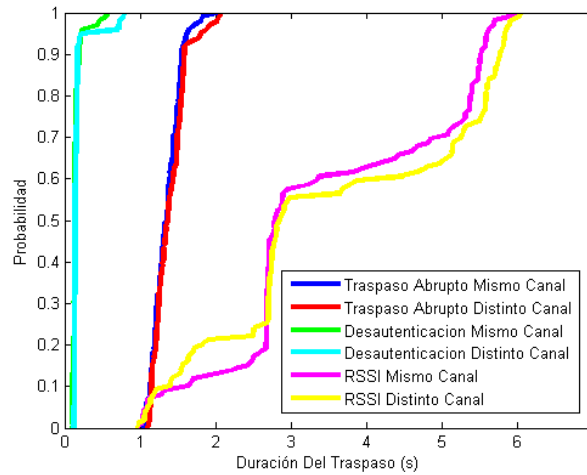


Figura 4.3: CDF de mecanismos de traspaso en Windows 8

4.2. Impacto del sistema operativo

Se ha visto en el apartado anterior que cada sistema operativo reacciona de manera diferente a cada mecanismo de traspaso. En esta sección vamos a hacer una comparativa de los distintos mecanismos de traspaso y su efecto en los diferentes sistemas operativos. Iniciaremos esta descripción observando la gráfica 4.4, que compara la reacción de los nodos móviles a un traspaso por desconexión abrupta del punto de acceso.

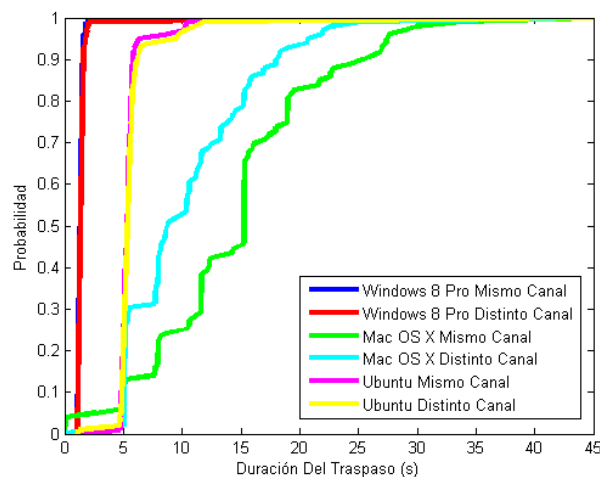


Figura 4.4: CDF del traspaso por traspaso abrupto

Viendo la gráfica 4.4 se concluye que esta clase de traspaso resulta óptima si el usuario se conecta a la red con un nodo que funcione con Windows 8 o Ubuntu. Por un lado, Mac OS X no ha obtenido buenos resultados a la hora de realizar las pruebas para este tipo

de traspaso. Una de las razones puede ser que, como comentamos en la sección anterior, el ordenador no realiza un escaneo activo y continuamente manda tramas de ahorro de potencia al punto de acceso. Por el otro lado, el network manager de Windows 8 ha conseguido terminar el 90 % de las pruebas en un tiempo máximo de 2 segundos. En el caso de Ubuntu, este percentil se garantiza a los 5 segundos.

A continuación, mostramos gráfica 4.5, en la cual podemos observar la CDF obtenida en los experimentos en los que se ha reducido la potencia del router al que el usuario se conecta. Podemos comprobar que sistema operativo que mejor ha reaccionado a esta clase de traspaso es Windows 8 ya que prácticamente en la totalidad de los casos se ha realizado el traspaso en menos de 5 segundos. El sistema que ha nos ha proporcionado peores resultados ante este mecanismo es Ubuntu, donde la duración máxima del 20 % de las pruebas es de 5 segundos. Por último Mac OS X alcanza el percentil 35 al realizar los traspasos en menos de 5 segundos. Una posible causa de estos resultados puede ser la persistencia de los network manager de mantenerse conectados al punto de acceso del que son desplazados.

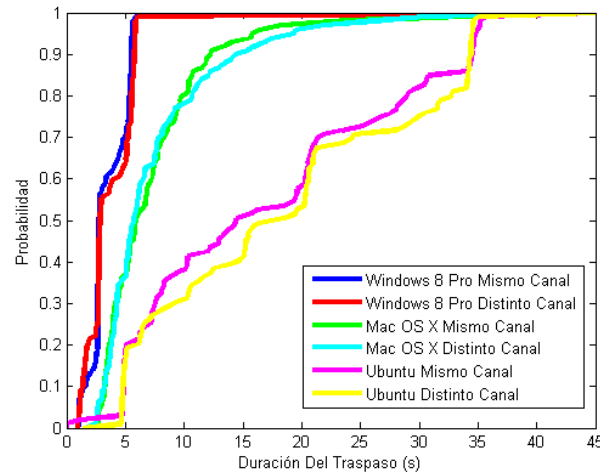


Figura 4.5: CDF del traspaso por RSSI

Creemos que es importante resaltar la similitud de las gráficas de Ubuntu y Mac OS X, teniendo la primera un crecimiento más suave que la segunda. A pesar de que la probabilidad de terminar en el mínimo tiempo posible para ambas es muy reducida, pasado más tiempo, la probabilidad de reconexión de Ubuntu siempre será menor que la de Mac usando este mecanismo. Por último, nos gustaría señalar que para ambas gráficas, todas las pruebas ha finalizado antes de 35 segundos, lo cual no es en absoluto eficaz.

Por último, queda interpretar la gráfica del método más rápido de realizar un traspaso: el envío de una trama de desautenticación por parte del punto de acceso. Este método ha resultado ser el más eficaz, tal y como puede comprobarse en la gráfica 4.6 que representa la CDF de las pruebas realizadas. Una de las ventajas que tiene este método es que solo afecta al nodo móvil al que queremos modificar su punto de acceso dentro de nuestra red, por tanto, es un mecanismo muy eficiente y sencillo. Sin embargo, no es escalable ya que el número de tramas de desautenticación que tenemos que mandar es directamente proporcional al número de nodos conectados al punto de acceso.

Como podemos comprobar en la gráfica, la probabilidad de que un nodo que funcione con cualquiera de los tres sistemas operativos probados realice su traspaso antes de 5

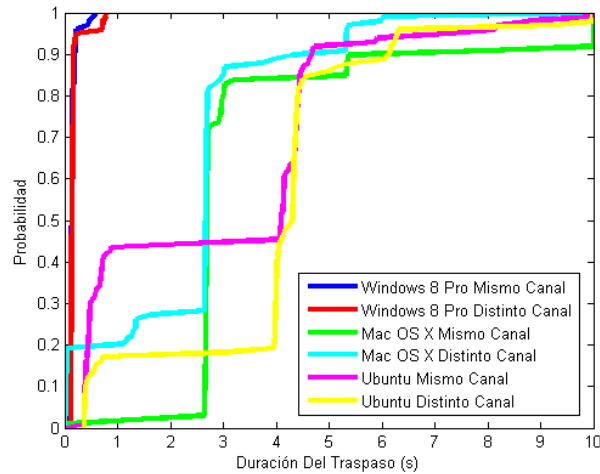


Figura 4.6: CDF del traspaso por desautenticación

segundos ronda el 90 %, siendo de nuevo el más eficiente Windows 8, cuya probabilidad de realizar el traspaso antes de 1 segundo se aproxima al 100 %.

4.3. Análisis de los mecanismos de traspaso

Para comprender mejor los resultados explicados en las secciones 4.1 y 4.2 vamos a mostrar diferentes gráficas con los intercambios de paquetes que se producen durante la ejecución de los mecanismos de traspasos.

4.3.1. Mecanismo por traspaso abrupto

En la figura 4.7 vemos el intercambio de paquetes del traspaso abrupto en Mac OS X. Consideraremos que la gran duración del traspaso en este caso se debe a que el nodo móvil envía tramas continuamente dedicadas al ahorro de energía (dichas tramas sirven para indicar al punto de acceso que el ordenador no está en reposo o para que el router abandone dicho estado). En segundo lugar, el retardo aumenta ya que la búsqueda de nuevas redes se realiza mediante escaneo pasivo, es decir, el router escucha todas las tramas de señalización que mandan los routers que se encuentran a su alrededor, lo cual puede resultar menos efectivo que una búsqueda activa.

En las gráficas 4.8 y 4.9 podemos comparar la señalización de un sistema operativo a otro. Podemos comprobar que Ubuntu tiene un tiempo medio de traspaso de 5 segundos ya que el intercambio de paquetes es similar al de Windows con la diferencia de que Ubuntu escanea todas las redes que tiene a su alrededor cuando quiere realizar el traspaso. Por último, Windows 8 es el que menos paquetes manda y gracias a su escaneo activo es la distribución que menos tiempo tarda en hacer el traspaso mediante este método.

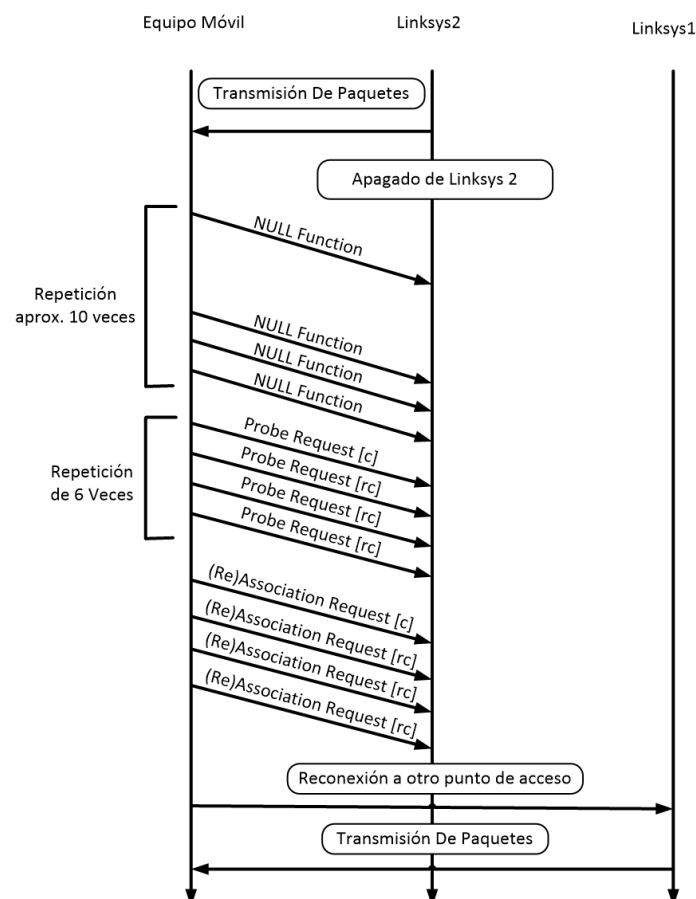


Figura 4.7: Intercambio de paquetes durante un traspaso abrupto en Mac OS X

4.3.2. Mecanismo de desautenticación

En la gráfica 4.10 podemos comprobar como Mac OS X intenta conectar de nuevo al punto de acceso del que ha sido desautenticado, esto prueba que su network manager puede haber sido programado para ser persistente con el punto de acceso al que se encuentra conectado. Una vez que ha recibido el paquete de desautenticación del router al que está conectado, envía dos tramas de autenticación al router del que ha sido expulsado. Tras esto inicia la conexión al otro router del entorno que difunde el mismo SSID.

Sin embargo, Ubuntu y Windows 8, antes de intentar reconectar a otro router de mismo SSID realizan un escaneo activo de los routers que tienen a su alrededor. Otra diferencia es que, mientras que Windows 8 interpreta que ha sido expulsado de un router, Ubuntu intenta conectarse a cualquier router que difunda el SSID al que está asociado, por tanto, manda tramas de autenticación tanto al router del que ha sido expulsado como al router al que va a conectarse.

4.3.3. Mecanismo de reducción de potencia

En la gráfica 4.13 podemos ver el intercambio de paquetes que realiza Mac OS X cuando se ejecuta el mecanismo de reducción de potencia. Como podemos ver, Mac OS X envía tramas RTS/CTS, cosa que los otros sistemas operativos no hacen.

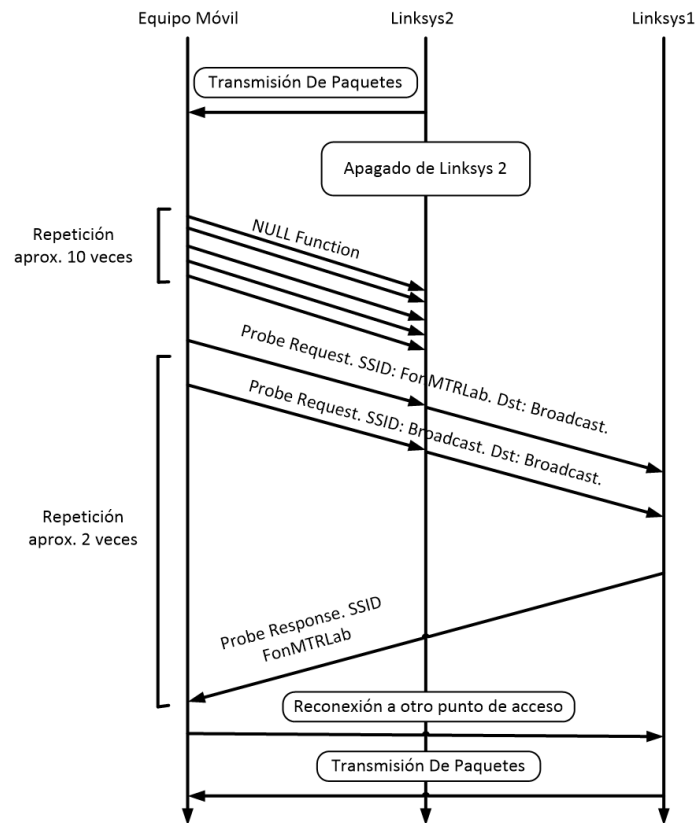


Figura 4.8: Intercambio de paquetes durante un traspaso abrupto del punto de acceso en Ubuntu

Mac OS X envía tramas RTS/CTS para evitar el problema del nodo oculto. Dicho problema parece que consiste en que cuando el nodo comprueba que la potencia que recibe se reduce, asume que puede haber otros nodos conectados al router. Por tanto, para evitar la colisión de tramas, el nodo reserva tiempo para transmitir mediante tramas RTS, las cuales son recibidas por el router.

En la gráfica 4.14 podemos comprobar el intercambio de paquetes durante un traspaso por reducción de potencia en Ubuntu. Si contrastamos la gráfica 4.5 con la mencionada antes, podemos concluir que el umbral de potencia de recepción de la tarjeta inalámbrica del nodo móvil es muy alta, con lo cual la duración del traspaso se incrementará debido a que la mínima potencia de transmisión del punto de acceso es suficiente para transmitir datos.

En las gráficas 4.14 y 4.15 podemos comprobar la distinta señalización que establecen los nodos con Mac OS X y Windows 8 conectados al router cuando detectan que la potencia de transmisión del router ha ido disminuyendo progresivamente. Se puede comprobar que Windows 8 y Ubuntu establecen una señalización más elemental mediante búsqueda activa al notar que la potencia transmitida ha sido reducida.

Al igual que el método anteriormente descrito de realizar traspasos, este método tiene una alta escalabilidad ya que influye en todos los nodos que están conectados al router sin necesidad de informar a cada nodo conectado. También puede ser útil en el supuesto que se quiera forzar la desconexión de los nodos que están más lejos de la antena de emisión puesto que los más cercanos podrán seguir recibiendo la transmisión sin que esto influya

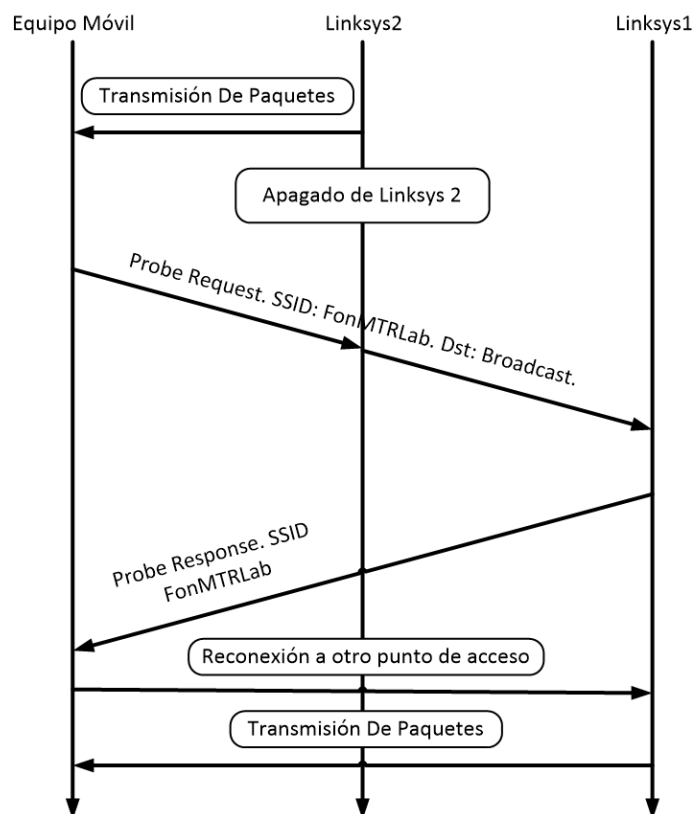


Figura 4.9: Intercambio de paquetes durante un traspaso abrupto del punto de acceso en Windows 8

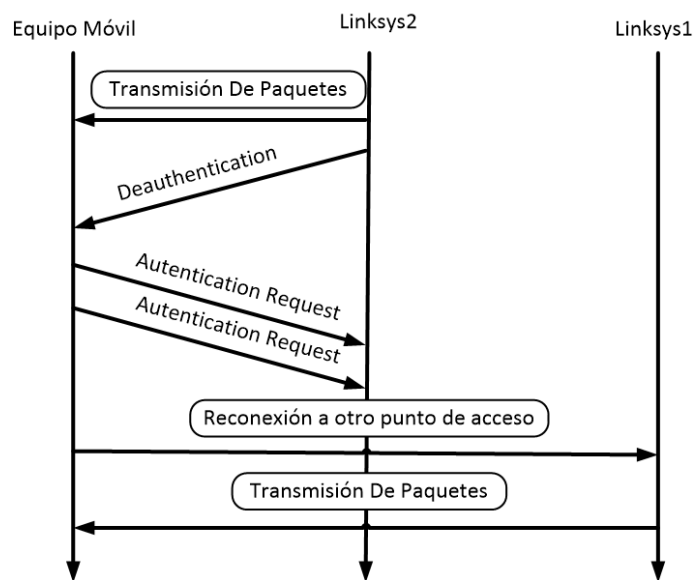


Figura 4.10: Intercambio de paquetes durante un traspaso por desautenticación en Mac OS X

en exceso en la calidad de la transmisión.

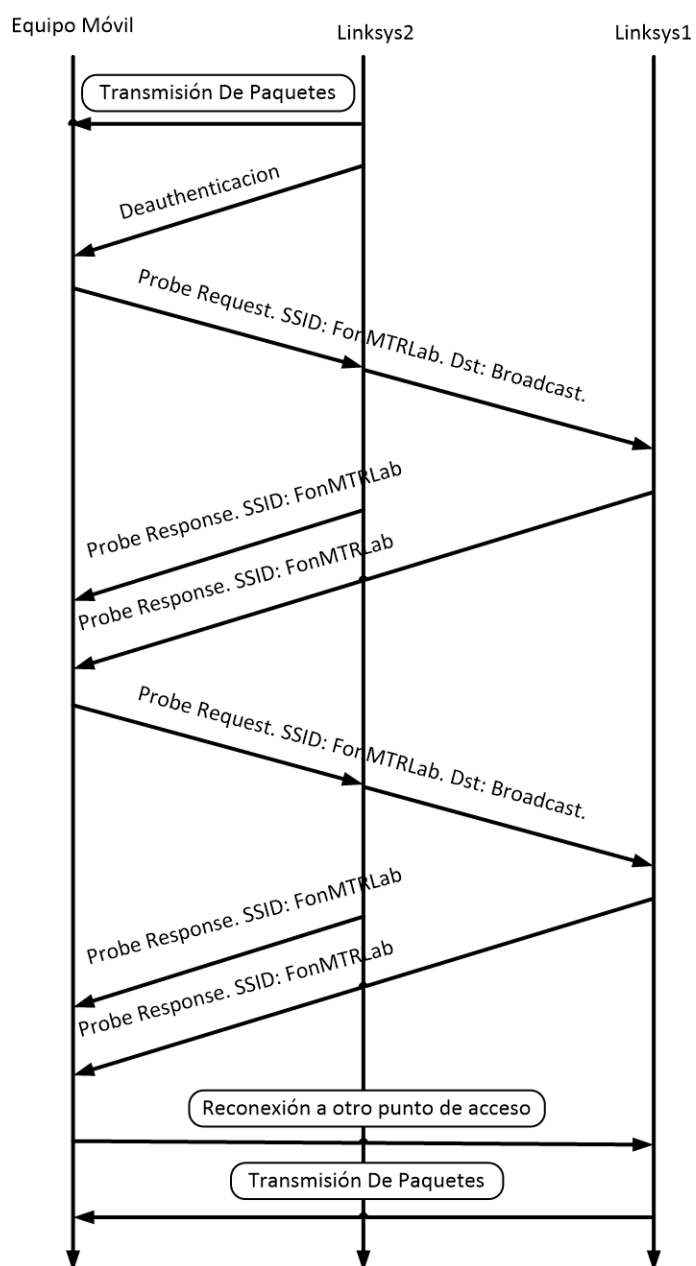


Figura 4.11: Intercambio de paquetes durante un traspaso por desautenticación en Windows

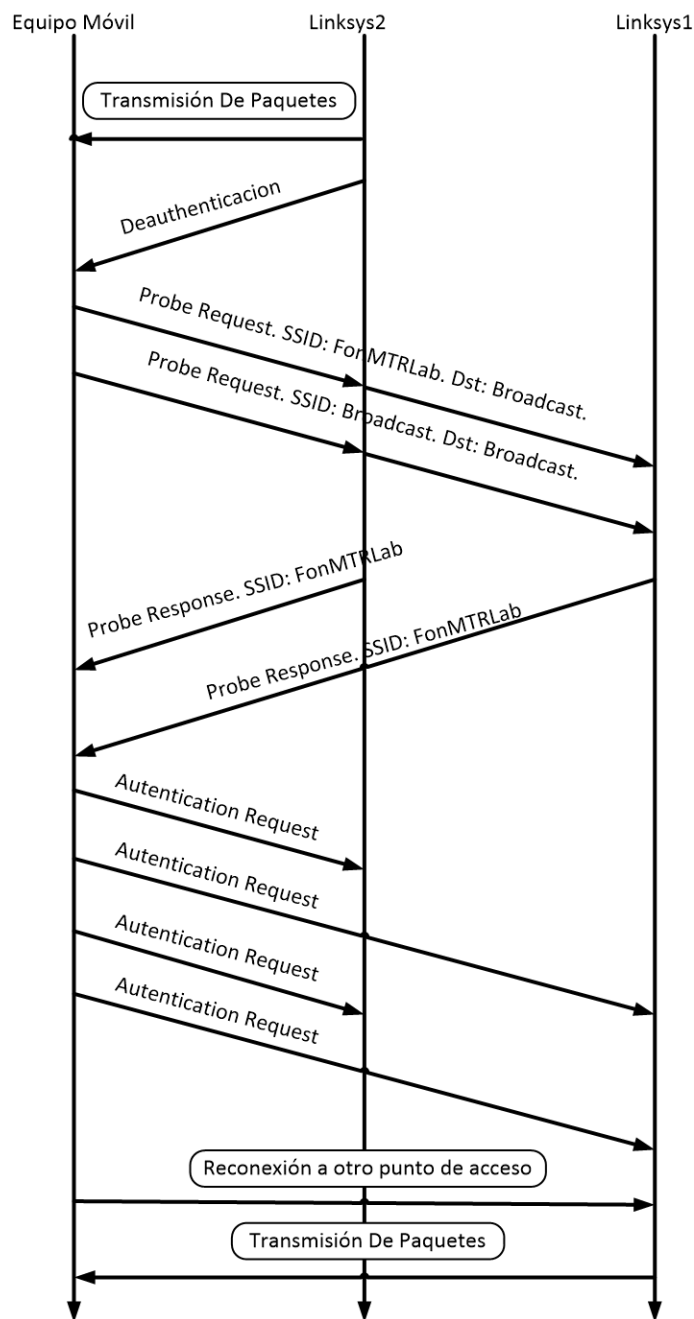


Figura 4.12: Intercambio de paquetes durante un traspaso por desautenticación en Ubuntu 12.10

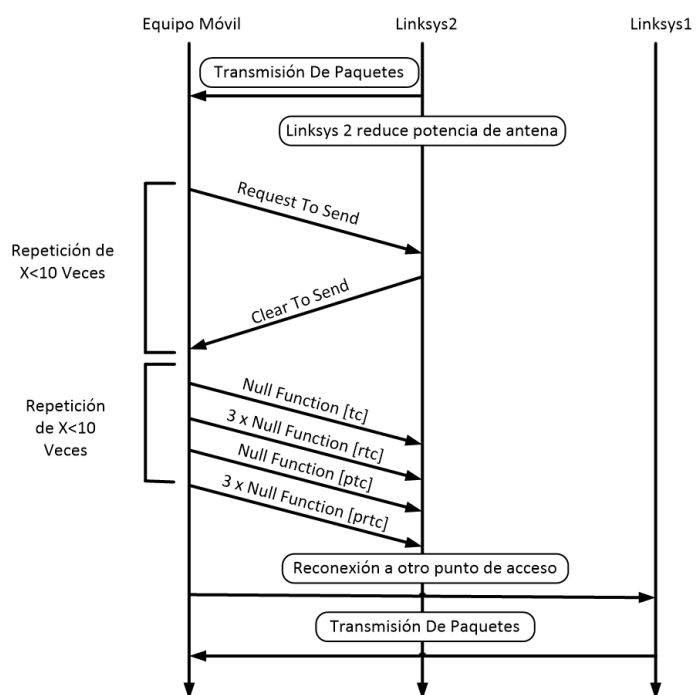


Figura 4.13: Intercambio de paquetes durante un traspaso por reducción de potencia en Mac OS X

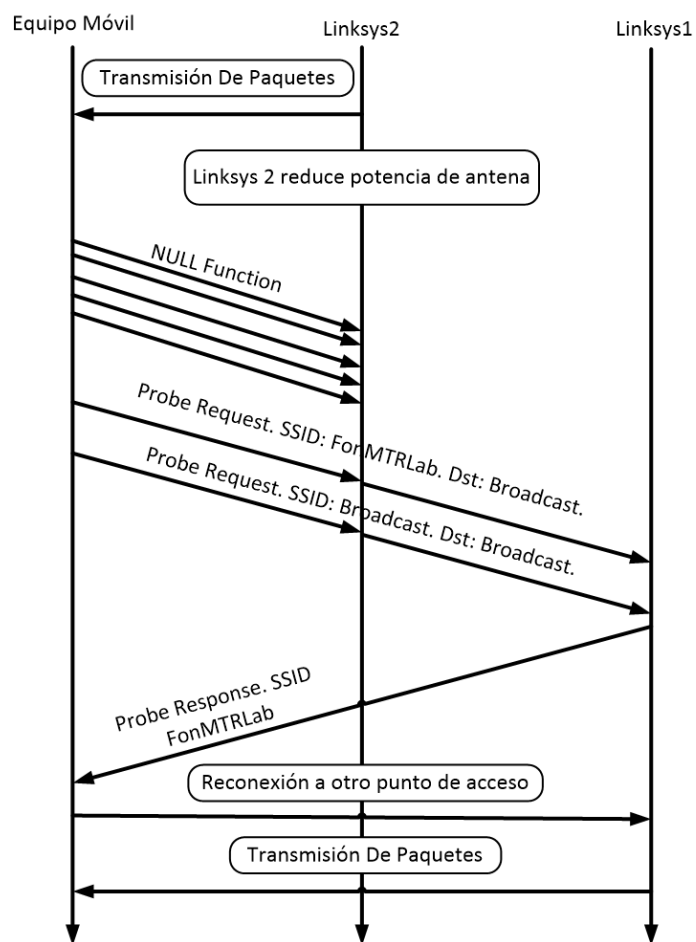


Figura 4.14: Intercambio de paquetes durante un traspaso por reducción de potencia en Ubuntu

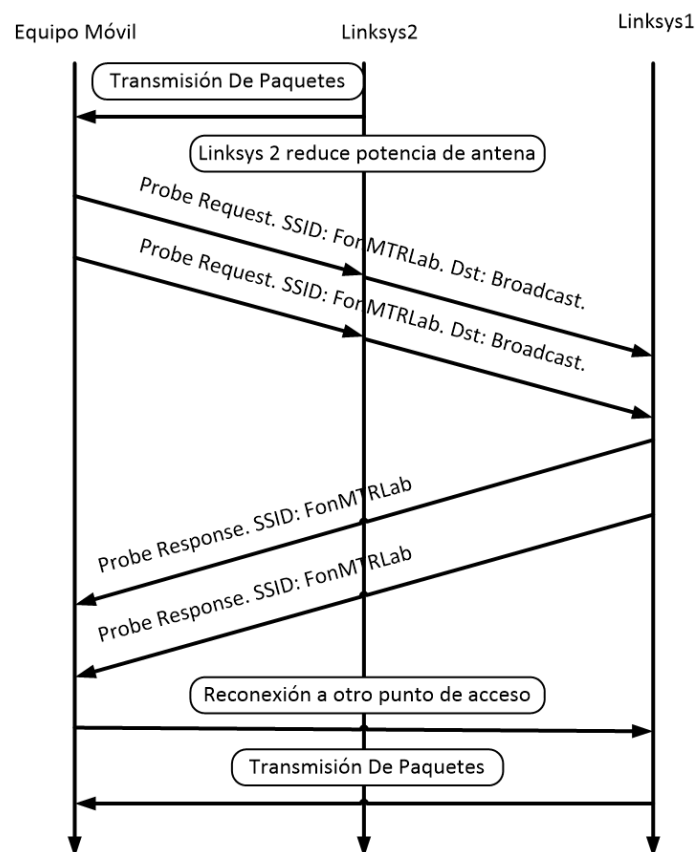


Figura 4.15: Intercambio de paquetes durante un traspaso por reducción de potencia en Windows 8

4.4. Redes con diferente SSID

Por último, vamos a hablar sobre el impacto de realizar un traspaso en condiciones similares a las anteriormente descritas con la salvedad de que los puntos de acceso difundirán un nombre de red (SSID) diferente. Para estas pruebas hemos decidido no realizar las pruebas en un entorno en el que los routers emitan en distinto canal ya que como hemos visto en las secciones 4.1 y 4.2 este hecho no influye considerablemente durante el transcurso de un traspaso. Para realizar estas pruebas, el network manager de los nodos móviles solo tiene almacenado en su memoria los SSIDs de las redes de pruebas a las que nos vamos a conectar. Se mantiene la misma IP en cualesquiera de los dos puntos de acceso, ya que cambiar rápidamente de dirección IP no es el objetivo de este estudio.

Sin embargo, los networks managers de los diferentes operativos no reaccionan adecuadamente ante esta clase de traspasos. En las pruebas anteriores hemos comprobado que los distintos sistemas operativos han reaccionado con mayor o menor rapidez ante un traspaso en el que los routers anuncian la misma red. Sin embargo, realizando pruebas de esta índole vemos que los ordenadores no realizan un correcto intercambio de paquetes para intentar asociarse a una nueva red.

Las únicas pruebas que hemos podido realizar con éxito han sido los traspasos por desautenticación y por desconexión abrupta del punto de acceso en el sistema operativo Windows 8, cuyo resultado pueden comprobarse en la figura 4.16. Como podemos comprobar, la probabilidad de realizar un traspaso antes de 15 segundos es del 90 % en el supuesto de un traspaso por desautenticación y de casi del 100 % en el caso de un traspaso abrupto. Los intercambios de paquetes son iguales a los de las gráficas 4.11 y 4.9, pero se realizan más lentamente debido al escaneo que se realiza al buscar otra red conocida de la lista del network manager.

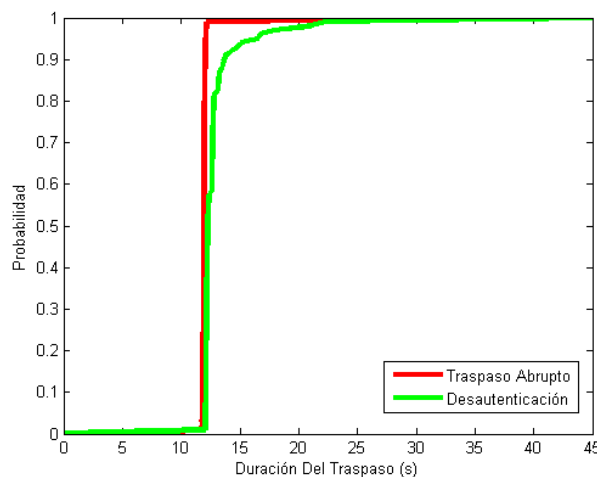


Figura 4.16: CDF del traspaso abrupto y por desautenticación en Windows 8 con los puntos de acceso en el mismo canal

Sin embargo, no ha sido posible ejecutar pruebas en los otros sistemas operativos. En el caso de Mac OS X, al no realizar escaneo activo, no busca activamente redes a su alrededor, y debido a esto no manda paquetes de autenticación a otras redes. En el caso de Ubuntu, la reacción a un traspaso que comprenda intercambio de redes es extremadamente lenta, del orden de 5 a 10 minutos.

4.5. Resumen

Por último, para intentar agrupar los resultados, mostramos en la siguiente tabla el tiempo máximo de un traspaso cuando se alcanza el percentil 90 en las pruebas que hemos realizado.

Sistema operativo Mecanismo de traspaso	Mac OS X		Ubuntu		Windows 8	
	MC	DC	MC	DC	MC	DC
Mismo Canal (MC) Diferente Canal (DC)						
Traspaso abrupto	25.0 s	16.0 s	5.1 s	5.2 s	1.3 s	1.3 s
Desautenticación	5.5 s	4.0 s	4.5 s	6.0 s	0.3 s	0.3 s
Reducción de potencia	13.0 s	16.0 s	35.0 s	35.0 s	5.1 s	5.2 s

A la vista de estos resultados podemos concluir que el sistema operativo que tarda menos tiempo en realizar los traspasos, sea cual sea el mecanismo, es Windows 8, destacando notablemente su rápida reacción a los traspasos por desautenticación, tanto en el mismo como en diferente canal. Los sistemas operativos basados en Linux también destacan por el traspaso por desautenticación, aunque no sea tan rápido como los que realiza Windows 8. En el caso de Ubuntu, hay que reseñar la escasa diferencia de tiempo que existe entre realizar un traspaso abrupto y uno por desautenticación (en caso de no importar la escalabilidad de la red uno sería buen sustituto del otro).

Esta tabla puede resultar valiosa para los puntos de acceso: a la hora de que un nodo quiera asociarse a un punto de acceso, dicho punto de acceso puede conocer el sistema operativo de un nodo por el rango de la dirección MAC con la que el nodo quiere registrarse en la red. En caso de que el punto de acceso tuviera demasiados nodos conectados, podría forzar a los nodos a conectarse a otro punto de acceso mediante el mecanismo más conveniente, es decir, el más eficaz en función de los tiempos que indica la tabla.

Capítulo 5

Conclusión y trabajos futuros

En esta memoria hemos podido comprobar cómo reacciona un nodo que funciona con diferentes sistemas operativos (Mac OS X, Ubuntu, Windows 8) a un traspaso entre puntos de acceso que funcionan con firmware Linux. Hemos clasificado los traspasos en función del canal en el que el router emite, del sistema operativo y del método de traspaso (abrupto, desautenticación, disminución de potencia). Después hemos realizado pruebas en función de los parámetros anteriores y en función del SSID que difunde el router.

Podemos comprobar que el sistema operativo que ha reaccionado con más rapidez a los distintos experimentos ha sido Windows 8, ya que en el 90 % de las pruebas, la duración máxima del traspaso ha sido de 5 segundos. No obstante, este estudio nos ha permitido comprobar la fiabilidad de los network manager de Mac OS X y Ubuntu, y por tanto, una buena continuación de esta memoria sería la implementación de mejoras en dichos network manager, de manera que permita reducir de manera considerable el tiempo de traspaso. Además, se podría repetir este estudio pero testando las principales plataformas de teléfonos móviles (Android, iOS, Windows Phone, etc) ya que la saturación de las redes de telefonía móvil empieza a ser un problema debido al gran volumen de datos que generan los smartphones.

También hemos podido estudiar que el mejor método para hacer un traspaso es que el router mande un mensaje de desautenticación al nodo al que desea cambiar de punto de acceso dentro de su red ya que es el método que menos impacto tiene en otros nodos que están conectados a un punto de acceso. Los métodos por desconexión abrupta y por reducción de la potencia de transmisión del punto de acceso influyen en todos los nodos que tienen asociados. Estos métodos pueden ser útiles cuando se desee hacer una gran cantidad de traspasos de muchos nodos para balancear la carga de la red. Un posible futuro trabajo puede ser la realización de las pruebas descritas en esta memoria en un entorno saturado (alta carga de datos) o el desarrollo de experimentos con el estándar 802.11v [HDS⁺10], el cual implementa numerosas mejoras en los mecanismos de señalización entre nodos y puntos de acceso.

Para reducir la duración de un traspaso, sería preciso desarrollar soluciones propietarias, tanto en el hardware del punto de acceso como en el del nodo móvil, puesto que las empresas desarrollan algoritmos de reconexión que funcionan mejor con hardware del mismo fabricante. Obviamente, esta idea es muy difícil de implementar puesto que muchos fabricantes que desarrollan puntos de acceso. Por tanto, parece lógico el buscar una solución teórica que satisfaga nuestra necesidad de una rápida reconexión. Para más información

sobre este tipo de traspaso, consulte [[MdIOS⁺07](#)].

Apéndice A

Planificación de tareas y presupuesto

A.1. Descomposición en tareas

En esta sección se va a presentar la descomposición de tareas que se han llevado a cabo para realizar este Trabajo Fin de Grado.

Se describen cada una de las tareas así como la relación con otras tareas y el esfuerzo de cada una.

- **Tarea A: Despliegue de la red inalámbrica**
 - **Subtarea A.1:** Estudio de los puntos de acceso y de los mecanismos de traspaso.
 - **Descripción:** en esta tarea se estudia el equipamiento que se va a usar, para conocer las características del mismo y comprobar que los mecanismos de traspaso se pueden ejecutar en el router.
 - **Objetivos:** conocer las características del equipamiento.
 - **Relación con otras tareas:** esta tarea da comienzo al proyecto.
 - **Duración:** 1 semana
 - **Recursos:** Ingeniero 0.25 ingenieros/mes.
 - **Subtarea A.2:** Instalación de OpenWRT en los nodos inalámbricos
 - **Descripción:** en esta tarea se instala la distribución OpenWRT 10.03.1 en los nodos inalámbricos Linksys WRT54GL y Fonera 2.0N.
 - **Objetivos:** se pretende que todos los nodo inalámbricos tengan una instalación personalizada de OpenWRT, que no dependa del fabricante de los nodos.
 - **Relación con otras tareas:** comienza después de la tarea A.1.
 - **Duración:** 1 semana
 - **Recursos:** Ingeniero 0.25 ingenieros/mes.
 - **Subtarea A.3:** Configurar el software en los nodos inalámbricos.
 - **Descripción:** en esta tarea se instalan todos los paquetes necesarios para el funcionamiento de los routers.
 - **Objetivos:** se pretende tener todos los nodos que se van a desplegar configurados correctamente.
 - **Relación con otras tareas:** esta tarea comienza después de la tarea A.2.

- **Duración:** 1 semana
- **Recursos:** Ingeniero 0.25 ingenieros/mes.
- **Subtarea A.4: Diseño de la red**
 - **Descripción:** se asignan las diferentes direcciones IP que va a tener cada nodo y el servidor, así como el routing necesario.
 - **Objetivos:** diseñar la red con las direcciones IPs necesarias.
 - **Relación con otras tareas:** esta tarea dará comienzo después de la tarea A.3.
 - **Duración:** 1 semana.
 - **Recursos:** Ingeniero 0.25 ingenieros/mes.
- **Subtarea A.5: Despliegue físico de los nodos**
 - **Descripción:** se colocan los nodos en los despachos, conectando todos los cables necesarios y se conectan las antenas apuntando a los nodos cercanos.
 - **Objetivos:** se espera tener la red desplegada para realizar las pruebas.
 - **Relación con otras tareas:** esta tarea comienza tras la tarea A.4.
 - **Duración:** 1 semana.
 - **Recursos:** Ingeniero 0.25 ingeniero/mes.
- **Tarea B: Documentación y análisis del estado del arte**
 - **Subtarea B.1: Estudio de los diferentes mecanismos de traspaso.**
 - **Descripción:** se estudian los diferentes mecanismos de traspaso que se pueden aplicar en redes 802.11 sin soporte del terminal.
 - **Relación con otras tareas:** esta tarea da comienzo tras la tarea A.
 - **Duración:** 1 semana
 - **Recursos:** Ingeniero 0.25 ingenieros/mes.
- **Tarea C: Desarrollo de código**
 - **Subtarea C.1: Desarrollo de scripts de automatización de medidas**
 - **Descripción:** durante este tarea se desarrolla un script en bash para medir el traspaso total.
 - **Objetivos:** tener un script que mide el traspaso total en la solución de nivel de enlace.
 - **Relación con otras tareas:** esta tarea comienza después de la tarea B.
 - **Duración:** 2 semanas.
 - **Recursos:** Ingeniero 0.5 ingenieros/mes.
- **Tarea D: Medidas**
 - **Subtarea D.1: Toma de medidas**
 - **Descripción:** en esta tarea se ejecutan los scripts desarrollados en todos los protocolos.
 - **Objetivos:** se espera tener medidos todos los tiempos del traspaso en todos los protocolos.
 - **Relación con otras tareas:** esta tarea da comienzo tras la tarea C.
 - **Duración:** 4 semana.
 - **Recursos:** Ingeniero 1 ingenieros/mes.

- **Tarea E: Resultados**

- **Tarea E.1:** Evaluación de los resultados

- **Descripción:** en esta tarea se estudia los resultados obtenidos y se dibujan las gráficas con el programa Matlab, que luego se usarán en la memoria.
 - **Objetivos:** analizar los datos y dibujar las figuras para la memoria.
 - **Relación con otras tareas:** esta tarea comienza tras la tarea D.
 - **Duración:** 1 semanas.
 - **Recursos:** Ingeniero 0.25 ingenieros/mes.

- **Tarea F: Memoria**

- **Subtarea F.1:** Organización y estructura de la memoria

- **Descripción:** se organiza la memoria y la estructura que va a tener.
 - **Objetivos:** organizar y estructurar la memoria.
 - **Relación con otras tareas:** esta tarea comienza tras la tarea E.
 - **Duración:** 1 semana.
 - **Recursos:** Ingeniero 0.25 ingenieros/mes.

- **Subtarea F.2:** Redacción de la memoria

- **Descripción:** durante esta tarea se redacta el documento.
 - **Objetivos:** redactar el documento con sus capítulos y anexos.
 - **Relación con otras tareas:** esta tarea da comienzo tras la tarea F.1.
 - **Duración:** 3 semanas.
 - **Recursos:** Ingeniero 0.75 ingenieros/mes.

- **Subtarea F.3:** Redacción del resumen

- **Descripción:** durante esta tarea se redacta el resumen de la memoria.
 - **Objetivos:** redactar el documento resumen de la memoria
 - **Relación con otras tareas:** esta tarea da comienzo tras la tarea H.2.
 - **Duración:** 1 semana.
 - **Recursos:** Ingeniero 0.25 ingenieros/mes.

Tarea	Duración (semanas)	Recursos (Ing/m)
Despliegue de la red inalámbrica		
A.1 Estudio de los puntos de acceso y de los mecanismos de traspaso	1	0.25
A.2 Instalación de OpenWRT en los nodos inalámbricos	1	0.25
A.3 Configurar el software en los nodos inalámbricos	1	0.25
A.4 Diseño de la red	1	0.25
A.5 Despliegue físico de los nodos	1	0.25
Total		1.25
Documentación y análisis del estado del arte		
B.1 Estudio de los diferentes mecanismos de traspaso	1	0.25
Total		0.25
Desarrollo de código		
C.1 Desarrollo de scripts de automatización de medidas	2	0.5
Total		0.5
Medidas		
D.1 Toma de medidas	4	1
Total		1
Resultados		
E.1 Evaluación de los resultados	1	0.25
Total		0.25
Memoria		
F.1 Organización y estructura de la memoria	1	0.25
F.2 Redacción de la memoria	3	0.75
F.3 Redacción del resumen	1	0.25
Total		1.25
Total		4.5

Tabla A.1: Resumen descomposición en tareas

A.2. Planificación con el diagrama de fases de ejecución detallado

En la figura [A.1](#) se presenta el diagrama de Gantt con todas las tareas que se han realizado en este Trabajo Fin de Grado.

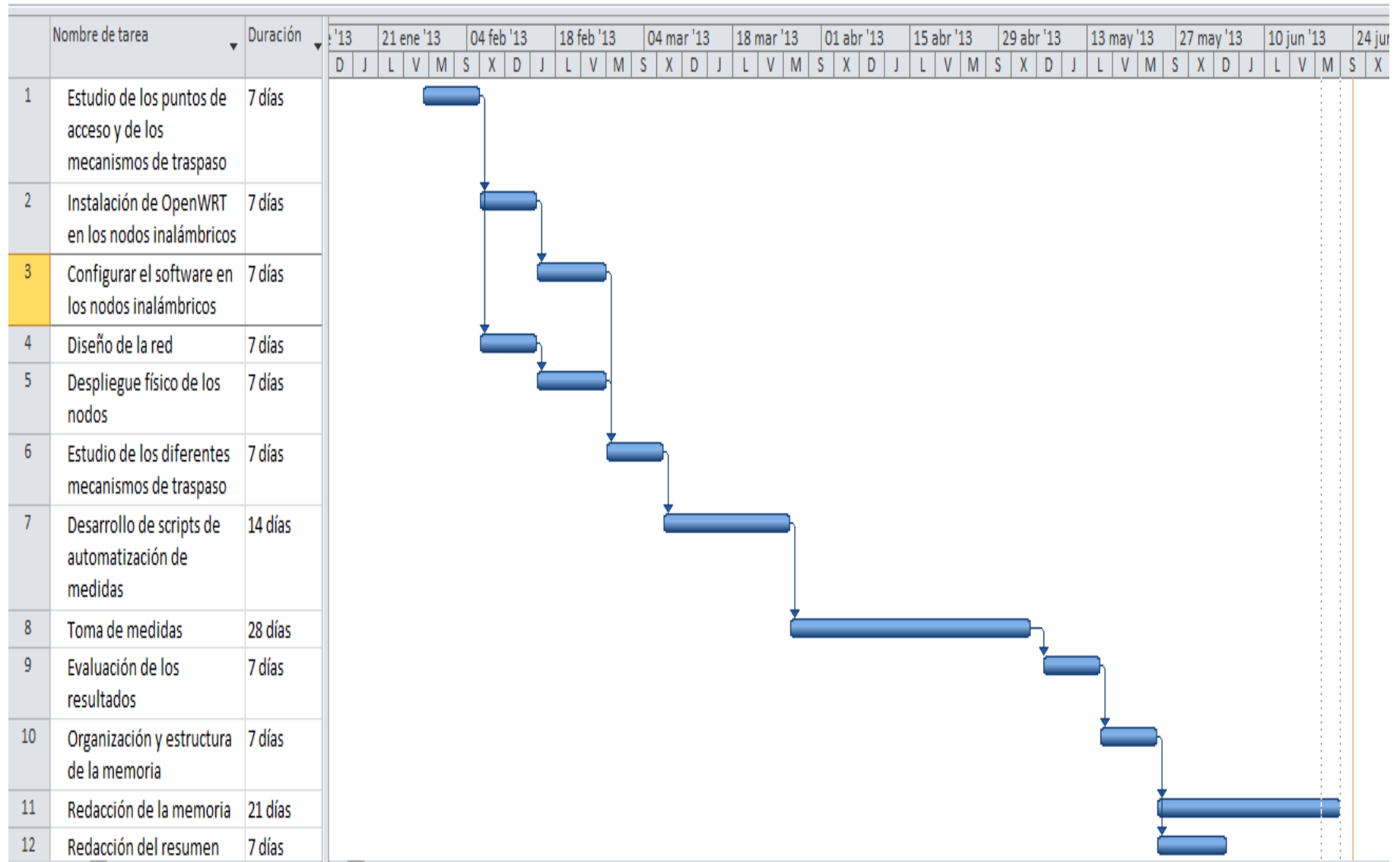


Figura A.1: Diagrama de Gantt del proyecto

A.3. Recursos

En esta sección se distinguen los diferentes recursos usados para la realización del Trabajo Fin de Grado:

- Recursos materiales:
 - Ordenadores de sobremesa.
 - 1 PC *Intel*[®] *Core*TM i7-2630QM @ 2.00GHz, 4GB RAM, Sistema Operativo: Windows 8 Profesional y Ubuntu 12.10
 - 1 PC *Intel*[®] *Core*TM 2 Duo @ 2.20GHz, 4GB RAM, Sistema Operativo: Mac OS X 10.6.8
 - 1 PC *Intel*[®] *Core*TM 2 Duo @ 2.66GHz, 2GB RAM, Sistema Operativo: Ubuntu 10.10
 - 2 Routers Linksys WRT54GL (incluidos cables de red y antenas).
 - 1 Router Fonera 2.0N (incluidos cables de red y antenas).
- Recursos de trabajo:
 - 1 Graduado en Ingeniería Telemática: 4.5 ingenieros/mes
 - 2 Ingenieros Senior: 0.5 ingenieros/mes
- Otros costes:
 - Conexión a internet durante 4.5 meses.

A.4. Presupuesto de Proyecto

1. Autor: Rubén Montoro Terciado
2. Departamento: Ingeniería Telemática
3. Descripción del Proyecto:
 - Título: Estudio experimental de mecanismos de traspaso controlados por la red en entornos 802.11.
 - Duración: 4.5 meses
 - Tasa de costes indirectos: 20 %.
4. Presupuesto total del Proyecto (valorado en Euros): euros. Ver tabla [A.2](#)
5. Subcontratación de tareas: no se especifican.
6. Otros costes directos del proyecto: no se especifican.

Concepto	Cantidad (€)	Coste €	% Proyecto	Dedicación (meses)	Depreciación (meses)	Total €
Recursos materiales						
Ordenadores de sobremesa	3	600	100	4.5	60	135,5
Router inalámbricos	3	60	100	4.5	60	13,5
Material Fungible	-	50	100	4.5	-	50
Total						198,50
Recursos de trabajo						
Graduado en Ing. Telemática	1 (4.5 ing/mes)	1523,39	-	-	-	6855.25
Ingenieros Senior	2 (0.5 ing/mes)	4.289,54	-	-	-	2.144,77
Total						9.000,02
Otros costes						
Conexión a internet	1	30	-	4.5	-	135
Total						135,00
Total Costes Directos						9333,52
Costes Indirectos						
Coste Indirecto	20 %	-	-	-	-	1.866,70
Total						1.866,70
Total Coste Proyecto						11.200,22

Tabla A.2: Tabla presupuesto

Apéndice B

Instalación y configuración de los routers inalámbricos

En este apéndice vamos a explicar como instalar el firmware OpenWrt en los routers utilizados en este Trabajo de Fin de Grado. Podemos consultar más información sobre dicho Firmware en [\[Teab\]](#)

B.1. Instalación de OpenWRT en Fonera 2.0N

1. Descargamos la versión de desarrolladores de firmware original de la Fonera en [\[Del12a\]](#). Esta versión del firmware habilita el acceso por SSH al router.
2. Conectamos cualquier interfaz ethernet del router a nuestro ordenador y configuramos una dirección del rango 192.168.10.0/24 en nuestro ordenador

```
sudo ip addr add 192.168.10.2/24 dev eth1
```

3. Accedemos a la configuración a través de un navegador web en la dirección 192.168.10.1 y accedemos al apartado de actualizaciones de firmware del router.
4. Actualizamos el paquete con el firmware para desarrolladores de Fonera [B.1](#) y esperamos a que el proceso se complete. El proceso dura aproximadamente 5 minutos.
5. Accedemos por SSH al router y descargamos el paquete OpenWrt para este router y lo instalamos [B.2](#)

```
cd /tmp
wget http://www.eriko.it/download/fonera/openwrt/
    openwrt-ramips-rt305x-fonera20n-squashfs-
    sysupgrade.bin

mv openwrt-ramips-rt305x-fonera20n-squashfs-
    sysupgrade.bin fon20nopenwrt.bin
mtd -r write fon20nopenwrt.bin image
```

6. Tras ejecutar las instrucciones anteriores el router se reinicia. Debemos configurar una nueva dirección IP del rango 192.168.1.0/24 en nuestro ordenador.



Figura B.1: Sección de actualización del router

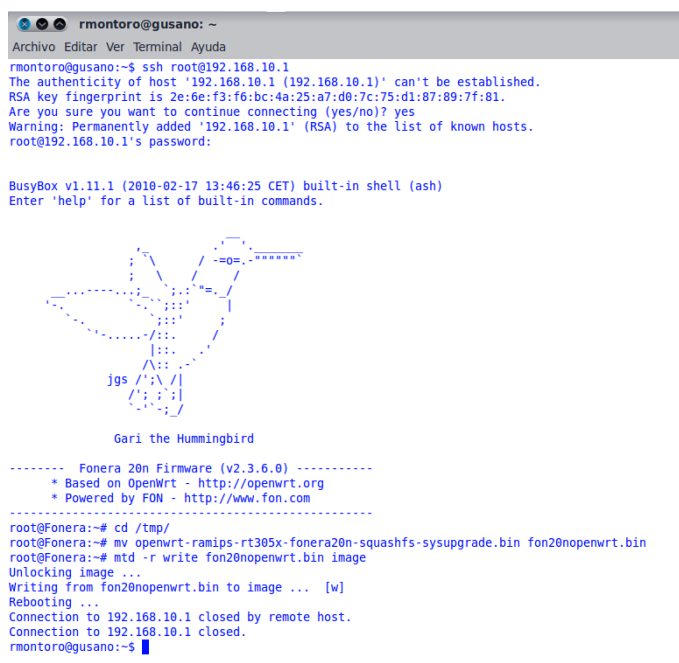


Figura B.2: Instalación de firmware OpenWRT por comandos

```
sudo ip addr del 192.168.10.2/24 dev eth1
sudo ip addr add 192.168.1.2/24 dev eth1
```

- Podremos acceder por Telnet al router o bien mediante su interfaz web en la dirección 192.168.1.1 para configurar nuestro router [B.3](#).

```
telnet 192.168.1.1
```

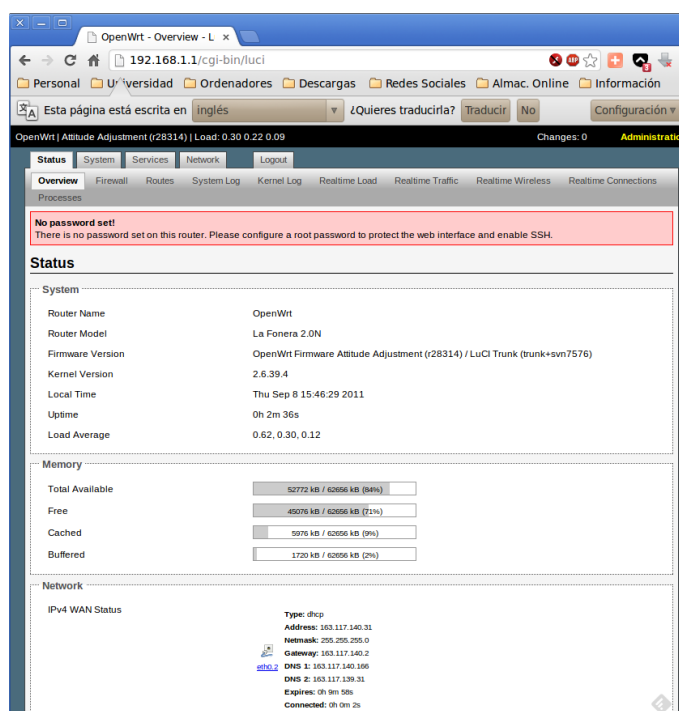



Figura B.3: Página principal del menú de configuraciones de OpenWrt del router Fonera 2.0N

8. Actualizar el router a la última versión del firmware (si es posible).

B.2. Instalación de OpenWRT en Linksys WRT54GL

1. Descargamos el firmware OpenWRT para Linksys WRT54GL desde [Teaa](#)
2. Conectamos el router a una interfaz ethernet de nuestro ordenador y configuramos una dirección del rango 192.168.1.0/24 en la interfaz que utilicemos

```
sudo ip addr add 192.168.1.2/24 dev eth1
```

3. Transferimos el archivo descargado en el paso 1 al router mediante el siguiente comando:

```
atftp --trace --option "timeout_1" --option "mode_octet" --  
put --local-file openwrt-wrt54g-squashfs.bin 192.168.1.1
```

Tras este paso, el router instalará OpenWrt en su memoria y se reiniciará automáticamente.

4. Accedemos al router por SSH y actualizamos el firmware a la última versión disponible [B.4](#). Dicha versión funciona con el kernel 2.4, el cual permite realizar modificaciones en la configuración inalámbrica sin sufrir problemas durante el funcionamiento.

```
ssh root@192.168.1.1  
cd /tmp/
```

```
wget http://downloads.openwrt.org/backfire/10.03.1/brcm-2.4/
openwrt-brcm-2.4-squashfs.trx
mtd write /tmp/openwrt-brcm-2.4-squashfs.trx linux && reboot
```

[illegible]

Figura B.4: Actualización de OpenWRT a la versión con kernel 2.4

5. Ejecutamos los siguientes comandos para poder restaurar el router en caso de fallo del sistema

```
nvrnm set boot_wait=on
nvrnm set boot_time=10
nvrnm commit && reboot
```

6. Accedemos a la interfaz gráfica desde el navegador web (192.168.1.1) para configurar el router a gusto del usuario [B.5](#)

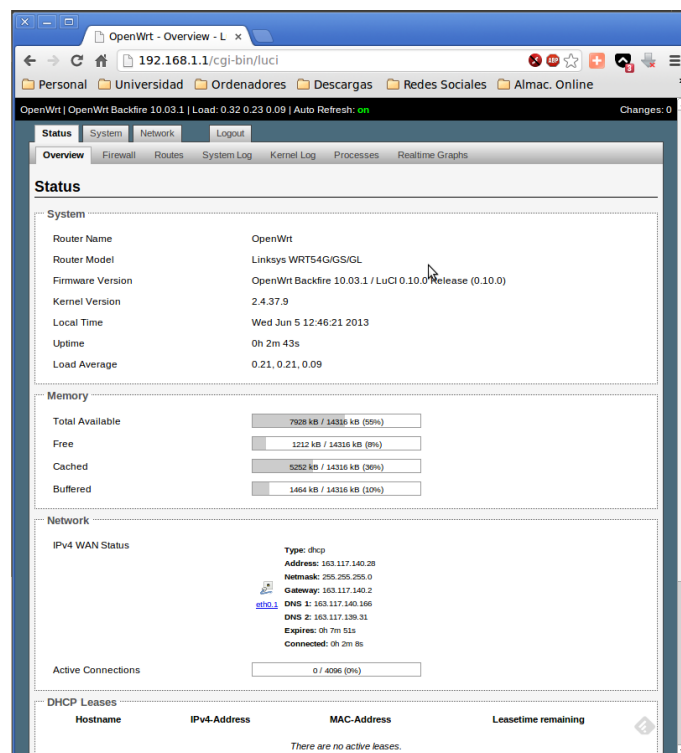


Figura B.5: Página principal del menú de configuraciones de OpenWrt del router Linksys WRT54GL

Bibliografía

- [ADH⁺06] Yair Amir, Claudiu Danilov, Michael Hilsdale, Raluca Musăloiu-Elefteri, and Nilo Rivera. Fast handoff for seamless wireless mesh networks. In *Proceedings of the 4th international conference on Mobile systems, applications and services*, MobiSys '06, pages 83–95, New York, NY, USA, 2006. ACM.
- [Cis13] Cisco. Cisco visual networking index: Forecast and methodology, from 2012 to 2017. *Visual Networking Index (VNI)*, may 2013.
- [Del12a] Enrico Deleo. Desarrollo de openwrt para fonera 2.0n. <http://erriko.it/internet-co/fonera/>, Agosto 2012.
- [Del12b] Enrico Deleo. Openwrt oficial fonera 2.0n + repositorio erriko.it [update r33265]. <http://erriko.it/internet-co/fonera/openwrt-ufficiale-per-fonera-2-0n-repository-erriko-it-update-r33265/>, Agosto 2012.
- [Gas05] Matthew S Gast. *802.11 Wireless Networks: The Definitive Guide, Second Edition*. O'Reilly Media, Inc., 2005.
- [HDS⁺10] Guido R. Hiertz, Dee Denteneer, Lothar Stibor, Yunpeng Zang, Xavier Pérez Costa, and Bernhard Walke. The iee 802.11 universe. *IEEE Communications Magazine*, 2010.
- [JPB⁺09] Amit P. Jardosh, Konstantina Papagiannaki, Elizabeth M. Belding, Kevin C. Almeroth, Gianluca Iannaccone, and Bapi Vinnakota. Green wlans: On-demand wlan infrastructures. *Mob. Netw. Appl.*, 14(6):798–814, December 2009.
- [Mal12] Dave Malpass. Cygwin install on windows 8 ssh-host-config as well! <http://www.davemalpass.com/words/cygwin-install-on-windows-8-ssh-host-config-as-well/>, Agosto 2012.
- [MdlOS⁺07] T. Melia, A. de la Oliva, I. Soto, P. Serrano, and R. Aguiar. Network controlled handovers: challenges and possibilities. *Springer*, pages 1 – 16, 2007.
- [MP06] Vivek Mhatre and Konstantina Papagiannaki. Using smart triggers for improved user performance in 802.11 wireless networks. In *Proceedings of the 4th international conference on Mobile systems, applications and services*, MobiSys '06, pages 246–259, New York, NY, USA, 2006. ACM.

- [MSA03] Arunesh Mishra, Minho Shin, and William Arbaugh. An empirical analysis of the ieee 802.11 mac layer handoff process. *SIGCOMM Comput. Commun. Rev.*, 33(2):93–102, April 2003.
- [RBPA07] Ramya Raghavendra, Elizabeth M. Belding, Konstantina Papagiannaki, and Kevin C. Almeroth. Understanding handoffs in large ieee 802.11 wireless networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, IMC '07, pages 333–338, New York, NY, USA, 2007. ACM.
- [Teaa] OpenWRT Developer Team. Linksys wrt54g, wrt54gl and wrt54gs. <http://wiki.openwrt.org/toh/linksys/wrt54g>.
- [Teab] OpenWRT Developer Team. Página principal de openwrt. <https://openwrt.org/>.